ROUTING AND ACTION

MEMORANDUM

ROUTING

TO:(1) Chemical Sciences Division (Becker, Jennifer)

Report is available for review

(2) Proposal Files Report No.: Proposal Number: 64617-CH.1

DESCRIPTION OF MATERIAL

CONTRACT OR GRANT NUMBER: W911NF-13-1-0367

INSTITUTION: Virginia Polytechnic Institute & State University

PRINCIPAL INVESTIGATOR: Amos Abbott

TYPE REPORT: Final Report

DATE RECEIVED: 9/15/14 9:04AM

PERIOD COVERED: 8/15/13 12:00AM through 3/14/14 12:00AM

TITLE: An Integrated System for Wildlife Sensing

ACTION TAKEN BY DIVISION

(x) Report has been reviewed for technical sufficiency and IS [x] IS NOT [ ] satisfactory.

(x) Material has been given an OPSEC review and it has been determined to be non sensitive and, except for manuscripts and progress reports, suitable for public release.

(x) Perfomance of the research effort was accomplished in a satisfactory manner and all other technical requirements have been fulfilled.

(x) Based upon my knowledge of the research project, I agree with the patent information disclosed.

Approved by SSL\JENNIFER.J.BECKER on 9/15/14 9:56AM

ARO FORM 36-E

REPORT DOCUMENTATION PAGE	2	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
15-09-2014	Final Report	15-Aug-2013 - 14-Mar-2014

4. TITLE AND SUBTITLE
An Integrated System for Wildlife Sensing

5a. CONTRACT NUMBER
W911NF-13-1-0367

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHORS
A. Lynn Abbott, Jeffry H. Reed

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Virginia Polytechnic Institute & State Unive
North End Center, Suite 4200
300 Turner Street, NW
Blacksburg, VA 24061 -0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
64617-CH.1

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This report describes the development of an integrated system for collecting visual data (video and still images) from an infrared camera along with registered sensor data (including GPS location) from an Android device. The camera is a FLIR BHM-Series product, and custom Android software was developed and tested using Nexus 4 and Nexus 5 phones. The user initiates a custom software app on the Android device, and then the user operates the camera as normal to acquire image and video files. After imaging operations have been completed, the user returns

15. SUBJECT TERMS
Infrared camera, FLIR, Android, Wildlife Management

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Amos Abbott
UU	UU	UU	UU		19b. TELEPHONE NUMBER
					540-231-4472

Report Title

An Integrated System for Wildlife Sensing

ABSTRACT

This report describes the development of an integrated system for collecting visual data (video and still images) from an infrared camera along with registered sensor data (including GPS location) from an Android device. The camera is a FLIR BHM-Series product, and custom Android software was developed and tested using Nexus 4 and Nexus 5 phones. The user initiates a custom software app on the Android device, and then the user operates the camera as normal to acquire image and video files. After imaging operations have been completed, the user returns to the Android app and selects commands that cause image and video files to be transferred from the camera to the Android device over a USB connection. The app will then synchronize the image and video files with sensor data from the Android, and will store the resulting data in an integrated format. This system can be used in low-light conditions for detection of wildlife, and therefore has the potential to improve the accuracy of wildlife population survey efforts.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

TOTAL:

Number of Manuscripts:

Books

Received Book

TOTAL:

Received Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Ahmed Ibrahim	1.00	
FTE Equivalent:	**1.00**	
Total Number:	**1**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
A. Lynn Abbott	0.05	
FTE Equivalent:	**0.05**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See Attachment

Technology Transfer

Report date:	**14 August 2014**
Report type:	**Final Progress Report**
Period covered by the report:	**15 August 2013 to 14 May 2014**
Title:	**An Integrated System for Wildlife Sensing**
Authors:	**A. Lynn Abbott and Jeffry H. Reed**
Organization:	**Virginia Polytechnic Institute and State University** **300 Turner Street NW, Suite 4200** **Blacksburg, Virginia 24061**
Contract number:	**W911NF-13-1-0367**

ABSTRACT:

This report describes an integrated imaging system that has been developed to support wildlife management operations. The system consists of an infrared camera (a FLIR BHM-6XR+) and a handheld Android device (an LG Nexus 4 E960). The camera was selected because many species of interest are nocturnal, requiring image collection in low-light conditions. The purpose of the Android device is to collect readings from its internal sensors, and to synchronize these readings as “metadata” with image and video files that are captured by the infrared camera. The Android-based sensors that are supported by this system are GPS position, compass direction, accelerometer, gyroscope, and barometer. The resulting integrated system can potentially be used to increase the accuracy and completeness of wildlife population survey efforts in support of U.S. military base wildlife management and conservation programs.

An Integrated System for Wildlife Sensing

Table of Contents

1. Problem Statement

The goal of this project has been to develop an integrated imaging system that can support wildlife management operations. Because many species of interest are nocturnal, an infrared camera will be used to collect thermal imagery in low-light conditions. Data-collection activities will be significantly enhanced if additional sensor data is collected along with the images and videos, particularly GPS (Global Positioning System) and compass direction. However, typical COTS infrared cameras do not contain these additional sensors. Because most cellphones and tablet computers do provide the additional data that is needed, the theme of this project has been to combine infrared imaging with sensor data collection from a separate handheld device. The resulting integrated system can potentially be used to increase the accuracy and completeness of wildlife population survey efforts in support of U.S. military base wildlife management and conservation programs.

The devices of particular interest to this project are the FLIR BHM-6XR+ infrared camera [1], and the LG Nexus 4 E960 cellphone running the Android 4.4 operating system. Both devices were selected and provided by the sponsor. This camera captures static images such as those shown in Figure 1, and is also capable of capturing video as illustrated in Figure 2. The Android-based phone can obtain readings from several additional sensors, including GPS. Researchers in this project have developed an integrated imaging system that synchronizes visual data from the camera with sensor data from the Android device.

The main issues that were addressed during this project are as follows. Each is described in the next section of this report:

Camera interfacing. The camera is a stand-alone product, and was not intended to be operated in a networked configuration. Several alternatives were considered for providing a USB-based communications capability between the camera and the Android device.

Android and USB. Today's phones are typically not configured to act as USB hosts. Some detective work was required so that this Android 4.4 device could act as a USB host to the FLIR camera.

Physical mounting. The Android device should be capable of being mounted directly onto the camera, but must remain detachable. Several alternatives were considered to satisfy this design requirement.

"Sensor Controller" software. A custom Sensor Controller application was developed for the Android device in order to collect and log readings from that device's sensors.

"Camera Controller" software. A custom Camera Controller application was developed for the Android device. This app transfers image and video files from the camera, and automatically synchronizes those files with sensor readings that have been logged by the Sensor Controller app.

Software development for this project utilized the latest Android SDK, which is available from [2].

Figure 1. Example images from the FLIR BHM-6XR+ infrared camera. These were captured soon after sunset with a 100-mm lens. Each image is 720 x 480 pixels in size, and is in JPEG format.

Figure 2. Example video frames from the FLIR BHM-6XR+ infrared camera. The video file is generated in AVI format, and each extracted image is 352 x 240 pixels in size. The green squares at the lower right of some of the frames indicate that the camera was undergoing an automatic recalibration.

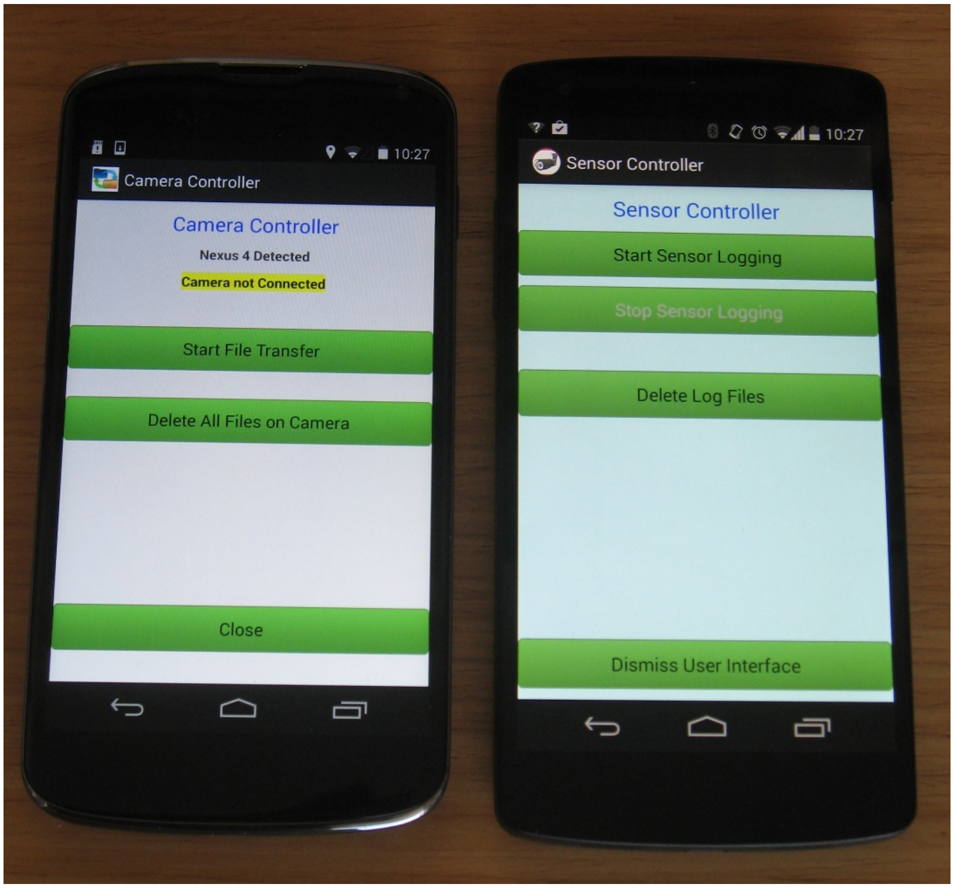
2. Summary of Most Important Results

2.1 Overview

A preliminary version of the integrated imaging system is shown in Figure 3. The primary components are the FLIR camera and the Android device, as introduced in the previous section. A quick-start guide for using the integrated system is provided in Appendix A.

Figure 3. The integrated imaging system, without the Android device mounted on the camera. The camera is shown with a 100-mm lens. The camera is on a tripod, although it can also be used in a handheld mode. The Android device displays the user interface for the custom Camera Controller application.

Software for the integrated imaging system has been partitioned into 2 separate Android applications (Figure 4). The Sensor Controller logs readings periodically from the Android device's organic sensors, and stores those readings into files for later use by the Camera Controller. After the user has acquired image and video files using the camera, the Camera Controller transfers those files from the camera to the Android device and then "synchronizes" the sensor data. Synchronization is performed by extracting sensor readings from the Sensor Controller's log files, and then creating new text files containing sensor readings for each image and video file that was transferred. These new text files contain sensor readings that begin 2 seconds before image/video acquisition, and continue for 2 seconds after image/video acquisition.



(a) (b)

Figure 4. Graphical interfaces to (a) the Camera Controller application, shown on a Nexus 4; and (b) the Sensor Controller, shown on a Nexus 5. Both of these custom Android apps were developed for this project. Both devices shown above are running the Android 4.4 operating system.

2.2 Camera Interfacing

This camera does not support networking, and does not support operation by remote control. A user operates the camera by pressing push-button controls that are located on top of the camera's housing. As the camera captures image and video data, it stores the data into files on a removable SD memory card. This SD card is accessible at the bottom of the camera, as shown in Figure 5. Also shown in the figure is a USB connector, which allows a host computer to access the SD card. Whenever an external device is connected to the camera's USB port, the camera does not respond to its pushbutton controls.

In this project, we considered 2 different approaches for interfacing the camera with the Android device. Initially we planned to utilize a WiFi-enabled SD memory card. The card would appear to the camera to be a standard SD memory card, but its WiFi capability would allow the Android device to access camera files using standard WiFi networking protocols.

Under guidance from the sponsor, however, we did not pursue the WiFi approach. Instead, we developed software that allows the Android device to become a USB host to the camera. When properly connected and initialized, the Android device sees the camera's SD card as a standard external flash memory. The Android device can then detect, transfer, and delete files that are on the camera's SD memory card while the card remains in the camera's SD slot. As mentioned above, *the camera does not respond to its pushbutton controls while a host is connected to the USB port.* Therefore, the USB connection between the Android device and the camera must be detached whenever images or video are being captured. After an imaging session has concluded, the user connects the Android device to the camera only briefly, using USB, in order to transfer image and video files from the camera to the Android device.

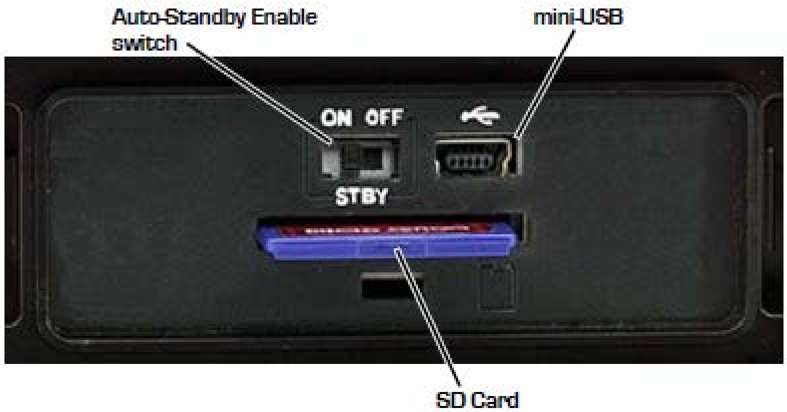


Figure 5. Interfaces to the FLIR camera, located at the bottom of the housing. Image and video files are stored by the camera on the SD card. The USB connector allows a host computer to access those files. The USB interface is also needed to allow a host computer to configure certain camera options, such as time of day.

2.3 Android and USB

It is not common for today's cellphones to act as USB hosts. Instead, the USB connection on a phone is typically used for charging the batteries, and occasionally for connecting to a laptop such that the phone is a USB client, not the host. As a result, we found that configuring an Android device to serve as a USB host for the FLIR camera was a harder task than anticipated.

The chosen Android device is a Nexus 4 phone, which we upgraded from Android version 4.3 to 4.4 during the project period. We later purchased a Nexus 5 phone for comparison. The Nexus 5 became available in November 2013, and it was delivered with Android 4.4 installed. For both the Nexus 4 and the Nexus 5, the Android 4.4 operating system must be "unlocked" and "rooted" in order to attain the privileged control that is needed to become a USB host. The procedure is described in [3]. The Nexus 4 requires the additional step of installing a custom kernel, which was obtained from [4].

Physical USB interconnections are described in Appendix B. When using the Nexus 4 as a USB host, an external power source is required. The Nexus 5 supports USB OTG, and therefore does not require the external power source.

2.4 Physical Mounting

In order to obtain meaningful sensor data, the Android device must be in close proximity and physically aligned with the camera during acquisition of images or video. The design specification calls for an ability to mount the Android device directly onto the camera, and the Android device must be detachable. As described previously, the USB cable must be disconnected while collecting images or video. The mounting arrangement must also take into account the tripod mount and the camera's relevant pushbutton controls.

After considering several alternatives for mounting the Android device onto the camera, 3 solutions are presented here. The first two solutions are simple, and make use of a cellphone case that allows the Android device to slide in and out easily. A third solution (not implemented) will require the fabrication of a physical mounting attachment.

All of these solutions require attaching the Android device to the underside of the camera. The underside is used because any attempt at mounting the Android device on the top of the camera body would block several pushbutton controls. Also, mounting the Android device on the side of the camera would be difficult because of curvature of the camera housing, and because one hand would be blocked by the Android device during hand-held usage.

Mounting option 1. The simplest solution involves a jogger's cellphone case. Figure 6 illustrates this solution. This cellphone case has a strap that wraps conveniently around the entire body of the camera, and is secured with velcro. Advantages of this method are that no permanent adhesive needs to be applied to the camera, and the entire case can be attached and detached quickly. A minor disadvantage is that 3 of the 5 pushbutton controls are covered by the strap. However, these 3 pushbuttons do not need to be changed often during the intended use of the system; the essential pushbuttons (for on/off control and for triggering image acquisition) remain clear. A bigger disadvantage is that the tripod mount is blocked, which means that this solution will typically require handheld operation. Another disadvantage is that the Android device can shift slightly relative to the camera, causing lower accuracy and repeatability in the metadata captured from the Android's sensors.

Mounting option 2. The second solution is illustrated schematically in Figure 7. For this option, a cellphone case is used again, but in a way that retains access to the camera's tripod mount. An adhesive can be used to attach the cellphone case to the battery door, on the bottom of the camera's housing. A minor drawback of this approach is the need for permanent adhesive to be applied to the camera housing. A larger disadvantage is that it may be difficult to mount the Android device every time with the same precise orientation on the camera.

Mounting option 3. A third solution is also indicated in Figure 7. This option, which has not been attempted, will require the fabrication of a new detachable door to replace the SD card door of the camera. The new attachment can be designed to contain a slot for the Android device, which should cause mounting to be precise and repeatable. The new attachment will cover the existing tripod mount of the camera, but a new tripod mount can be incorporated into the attachment.

(a) (b)

(c)

Figure 6. Mounting solution. A camera case with a strap secures the Android device to the underside of the camera. (a) Top view of camera. The strap covers 3 of the 5 pushbuttons, but the essential controls are not blocked. (b) Bottom view of camera. (c) The integrated system. This mounting method is convenient when a tripod is not needed.

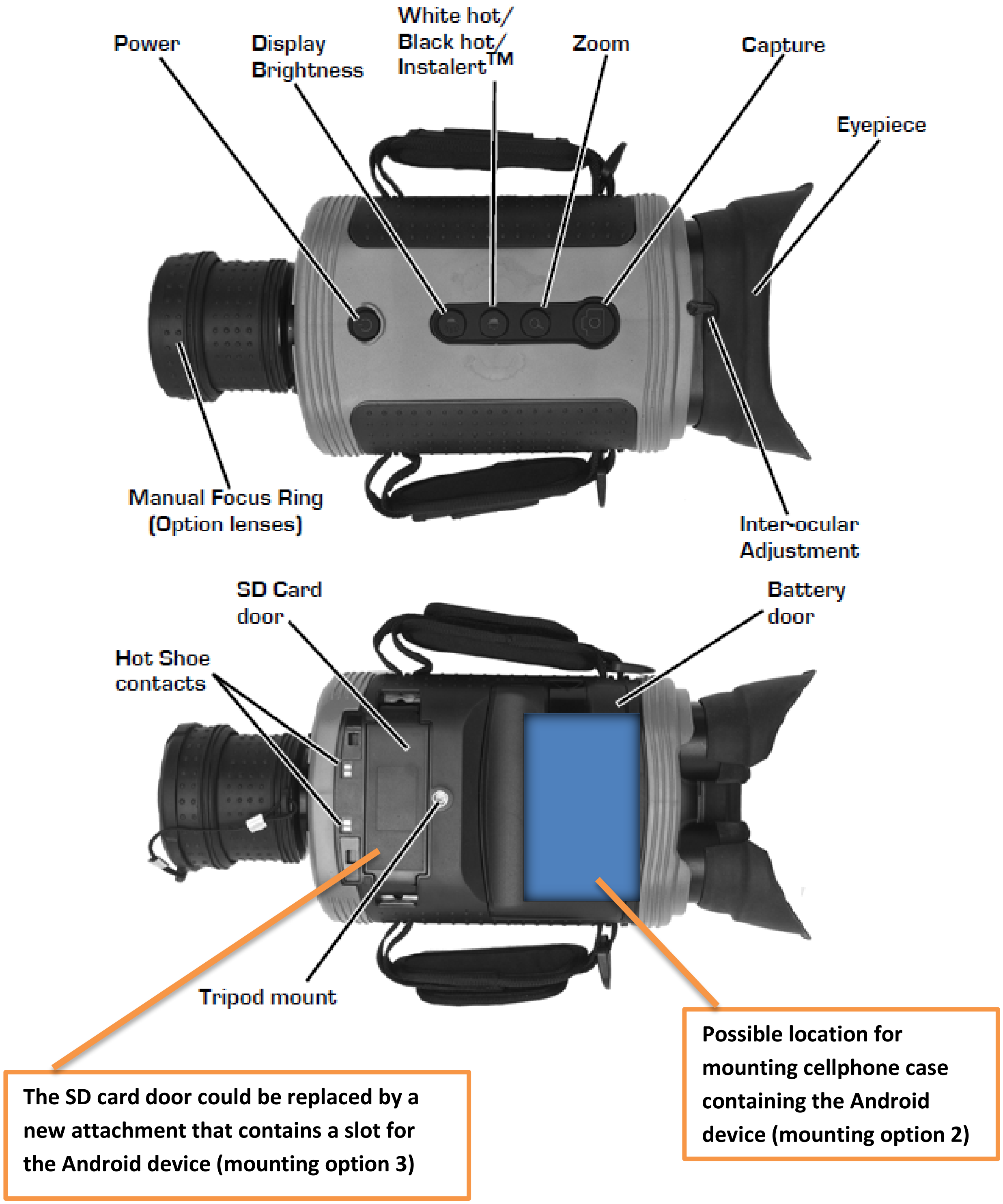


Figure 7. To allow for using a tripod, the Android device can be mounted as shown. *Option 2:* At the location shown in blue, a cellphone case can be attached using an adhesive and velcro. *Option 3:* For more precise mounting, new attachment to replace the SD memory card door could be fabricated. Both of these locations allow complete access to the 5 pushbuttons on top of the camera. (Figure adapted from [1].)

2.5 “Sensor Controller” Software

The Sensor Controller application obtains readings from several organic sensors on the Android device. The app does not start collecting data automatically. Instead, the user must direct the app to “Start Sensor Logging”, and then the app operates in the background and collects sensor data continuously until the user directs it to “Stop Sensor Logging”. (The graphical user interface for this app was shown in Figure 4(b).) Naturally, the app will also halt if the battery is drained or if the device is rebooted. However, the app does continue to read and log data when the device is in “sleep” mode.

Sensor data values are “logged” by storing them into separate text files, one file per sensor type. Each log file is a text file, and each line of text corresponds to one sensor reading. The format of those files for each sensor is given below. Over time, these log files can become large, so the user will need to “Delete Log Files” from time to time. The natural time to remove log files is right after synchronizing the sensor data, as described in the next section. The 16-GB Nexus 4 Android device has the storage capacity to log sensor data continuously for about 30 days. The actual amount of time depends in part on the size and number of files that are already in storage. The amount of time also depends on the amount of device movement. When the device is stationary, some of the sensors provide fewer readings, and therefore the log files may grow at a slower pace. Similarly, when GPS updates are not available, as may occur indoors, then updates are not sent to the GPS log file.

The last of the user controls is “Dismiss User Interface”. This control clears the screen, but it does not cause the app to stop collecting and logging sensor data.

The organic Android sensors that are accessed by this app are listed in Table 1, for both the Nexus 4 and the Nexus 5. The reader is referred to [5] and [6] for more details concerning sensors on Android devices. The following is a brief discussion of each sensor type.

Coordinate reference frame. The coordinate system for most Android sensors is illustrated in Figure 8. When the screen of the device is held in its default orientation (as shown in the figure), the x axis is horizontal and points to the right, the y axis is vertical and points upward, and the z axis points away from the screen toward the user. Locations behind the screen therefore have negative z values. The axes are not swapped when the device's screen orientation changes.

Table 1. Android sensors supported by the Sensor Controller and Camera Controller apps. In addition to the quantities in the table, each sensor reading includes a UTC timestamp. Sensor identifiers in the range 1 through 10 are consistent with Android identifiers. The Android API does not specify identifiers for GPS or compass.

Sensor type	*Sensor identifier*	*Measurement type*	*Units*
Accelerometer	1	Accelerations along x, y, and z axes (includes effect of gravity)	meters / second^2
Gyroscope	4	Rates of rotation about x, y, and z axes	radians / second
Barometer	6	Atmospheric pressure	millibars (hPa)
Linear accelerometer	10	Linear accelerations along x, y, and z axes (excludes effect of gravity)	meters / second^2
GPS	91	Latitude, longitude, altitude, and ground speed	degrees, degrees, meters, and meters / second
Compass	92	Orientation relative to magnetic north, plus pitch and roll angles	degrees

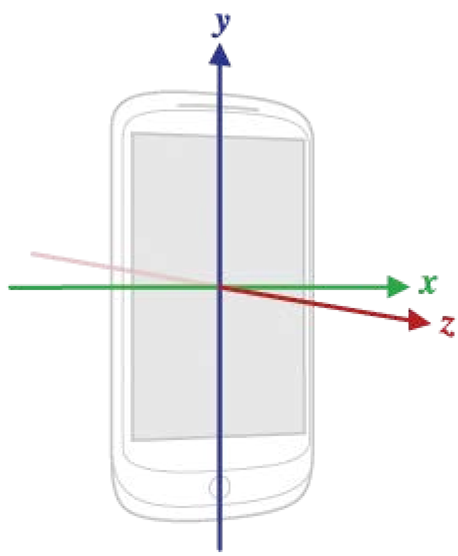


Figure 8. Coordinate reference frame for Android devices (from [5]).

Accelerometer (sensor identifier 1) and linear accelerometer (sensor identifier 10). The Android API distinguishes between the actual hardware accelerometer, which includes the effect of gravity, and a virtual “linear acceleration sensor,” in which the effect of gravity has been subtracted out. For convenience, both of these are provided by the Sensor Controller application. A single reading is one line of text in the log file, given as a comma-separated list with the following format:

Time stamp, acceleration_x, acceleration_y, acceleration_z,

As shown, each line of text in a log file ends with a comma. The time stamp is in the format "yy-mm-dd hh:mm:ss.sss". "Acceleration_x" refers to acceleration in the direction of the x axis, in units of meters/second^2. Likewise, "acceleration_y" and "acceleration_z" refer to acceleration in the directions of the y and z axes, respectively. For sensor identifier 1, gravity contributes 9.81 meters/second^2 in a direction that depends on the orientation of the device. For example, when the device is lying on a table the sensor reports a value of –9.81 in the z direction. The following is an example reading:

14-03-12 19:10:16.556,-0.17056835,2.1583471,0.73923755,

Gyroscope (sensor identifier 4). Each reading has the format

Time stamp, angular_x, angular_y, angular_z,

The time stamp is in the same format as before. "Angular_x" is the rotational speed about the x axis in radians/second, with positive values indicating counterclockwise rotation. Similarly, "angular_y" and "angular_z" represent rotational speeds about the y and z axes, respectively. The following is an example:

14-03-12 19:13:13.362,-0.012771606,0.038650513,-0.017196655,

Barometer (sensor identifier 6). Each reading has the following format:

Time stamp, pressure,

The time stamp is in the same format as described above. This sensor provides a measurement of atmospheric pressure, in units of millibars (hPa). An example reading is as follows:

14-03-12 19:13:21.089,928.7995,

GPS sensor (sensor identifier 91). The Android API provides GPS information through a "location provider," which is software that typically integrates information from GPS, WiFi routers, and cellphone towers. The Sensor Controller app disables input from WiFi and cell towers, and therefore provides GPS information only. A single reading is a comma-separated list with the following format:

Time stamp, latitude, longitude, altitude, ground speed,

The time stamp is in the same format as before. Latitude and longitude are given in degrees, altitude is given in meters, and ground speed is given in meters/second. The following is an actual reading that was obtained near Blacksburg, Virginia, and the values agree reasonably well with known geolocation data:

14-03-12 19:29:17.126,37.26519746,-80.39522701,622.0,0.5,

Compass (sensor identifier 92). The Android API provides compass-related information through a software "orientation function," which integrates information from the organic geomagnetic field sensor and the accelerometer. A single reading has the following format:

Time stamp, azimuth, pitch, roll,

Azimuth is the angle between the magnetic north direction and the y axis of the Android device. A sample reading is as follows:

14-04-23 10:16:46.220, 88.88398,9.850034,-90.662100,

2.6 "Camera Controller" Software

The primary purpose of the Camera Controller app is to transfer image and video files from the FLIR camera to the Android device, and then to "synchronize" those image and video files with organic Android sensor readings. Synchronization is performed by determining the acquisition time for each image/video file, and then creating a set of individual "sensor" files for each image/video file. Note that synchronization is possible only if the Sensor Controller app started logging sensor data at least 2 seconds *before* the camera was used to capture images or videos.

For this camera, the file name appears to be the most accurate way to get the capture time for any image or video. The manufacturer's naming convention for camera files is

DSC_ddmmmyy_hhmmss.ext

Example file names are

DSC_12Mar14_191025.jpg and MVI_12Mar14_191743.avi

This image file (extension "jpg") was captured on the 12th of March, 2014, 25 seconds after the hour 19:10 (equivalent to 7:10 p.m.). The video file (extension "avi") was captured several minutes later on the same day, starting 43 seconds after the hour 19:17.

For each image and video file, the Camera Controller app examines the file name to determine the date and time, and then generates synchronized text files with the following naming conventions:

file.syn	– contains one line stating date and time when synchronization was performed (also acts as an placeholder to prevent synchronization a second time)
file_1.txt	– contains accelerometer sensor readings synchronized with file.jpg or file.avi
file_4.txt	– contains gyroscope sensor readings synchronized with file.jpg or file.avi
file_6.txt	– contains barometer sensor readings synchronized with file.jpg or file.avi
file_10.txt	– contains "linear accelerometer" sensor readings (excludes effects of gravity)
file_91.txt	– contains GPS sensor readings synchronized with file.jpg or file.avi
file_92.txt	– contains compass sensor readings synchronized with file.jpg or file.avi

For example, the file DSC_12Mar14_191025_4.txt will contain gyroscope sensor readings for image file DSC_12Mar14_191025.jpg, beginning 2 seconds before and continuing 2 seconds after the image was captured. Similarly, file MVI_12Mar14_191743_91.txt will contain GPS readings for video file

MVI_12Mar14_191743.avi, beginning 2 seconds before the time indicated by the file name, and continuing until 2 seconds after the video ends.

Each of the sensor files contains readings in a comma-separated format, as described in the previous section, for convenience in importing the sensor data into utilities such as Excel.

When the Camera Controller app is started, it first checks the type of Android hardware that is running the app. It then reports “Nexus 4 Detected” or “Nexus 5 Detected”, if appropriate. A Nexus 4 is checked to see if it has been properly “batched,” and a Nexus 5 is checked to see if it has been properly “rooted”.

The app then automatically checks the device’s USB port, and reports “Camera Connected” or “Camera not Connected”. The camera can be attached while the app is running, and the new connection will be detected automatically.

If the user selects “Start File Transfer” when the camera is connected, then the app will begin to transfer *all* image and video files from the camera to the Android device. During this phase, the files are copied but not deleted from the camera’s SD memory card. After the files have been copied, the app automatically begins to perform the synchronization step that was described above. Status information is displayed during transfer and synchronization, which can take several seconds or several minutes, depending on the number of image and video files, and on the sizes of the log files.

When the user is satisfied that all relevant files have been transferred and synchronized, the user can select “Delete All Files on Camera”. This selection will cause the SD memory card in the FLIR camera to be completely erased.

The user can select “Close” to stop the Camera Controller app. At this time, all of the image and video files, with their respective synchronization files, are available on the Android device’s file system. The user is free to examine any of these file using local Android apps. The user can also transfer any of those files to a host computer for further analysis.

2.7 Setting Date and Time in the Camera

It is important to verify that the camera’s date and time settings are correct. This step is necessary for proper synchronization of image and video files with Android sensor data. Setting the date and time is not trivial, however.

For the FLIR BHM-6XR+ camera model, it appears that the date and time cannot be set from the pushbutton switches, but must be configured from a laptop through a USB connection. In our trials with a Windows 7 PC, drivers for the camera were not found and installed automatically when we connected the camera to the PC. Instead, we had to locate drivers for the camera manually and see that they were properly preconfigured on the laptop before making the USB connection.

2.8 Summary

The goal of this project has been to develop a tool to increase the accuracy and completeness of wildlife population survey efforts in support of U.S. military base wildlife management and conservation programs.

The result is a novel system that integrates an infrared camera with an Android device. A custom software application was developed to transfer image and video files from the camera to the Android device, and then supplement those camera files with synchronized sensor data from the Android device. The system has been tested with a Nexus 4 phone and a Nexus 5 phone, both running Android 4.4. By taking advantage of sensors on the Android device, the system should enable game wardens of military bases to develop and implement wildlife management plans that will increase safety for both wildlife and base personnel. The system should also enhance research and partnerships with local, state, and federal wildlife management agencies.

3. Possible future directions

This section outlines several opportunities for future work. Please note that the current set-up requires the Android device to be disconnected from the FLIR camera during image/video acquisition. This means that processing tasks cannot be performed in real-time, unless a different camera is selected.

1. Improved mounting. Section 2.4 described three options for mounting the Android device on the FLIR camera. The best option calls for fabricating a new attachment for the camera, but this has not been implemented. The attachment would replace the camera's existing SD card door, and would contain a slot for rigidly mounting the Android device. This mounting arrangement would make it possible to align the Android device with the camera much more precisely, and would accommodate a tripod as well.

2. Multisensor fusion. Additional sensors could be mated with the FLIR camera, particularly range sensors. Examples include light-weight laser rangefinders and ultrasound devices. In some cases, multisensor fusion can be performed using information theoretic techniques.

3. RF direction finding. In addition to range estimation, novel approaches to direction estimation are possible beyond the organic sensors on the Android device. In particular, RF-based approaches could be used.

4. Fusion with Google Maps. Using geolocation and compass information from the Android device, it should be possible to create an interactive display that shows the camera position and view direction superimposed on a Google Maps display. Because lens parameters are known, it should also be possible to highlight the camera's field of view on the map. The information could be generated for a single camera or for multiple cameras, at a single point in time or over specified duration. Such information should be useful in planning future camera placement.

5. Selective filtering of IR images. Low-level analysis can be performed on the Android device to identify potential objects and situations of interest. Image regions can be detected based on predetermined intensity levels and object sizes within the image. Simple shapes could be tracked, and direction and speed within the image can be computed. Some of these tasks might benefit from the open-source computer-vision software known as OpenCV, which is described at http://opencv.org/platforms/android.html. (Other tasks in this list may also benefit from OpenCV.)

6. Object recognition. Mid-level recognition tasks could be performed on the Android device, and higher-level analysis could be performed on a larger server. Based on 2D shape, for example, an object might be classified as one of the following: {animal, vehicle, person}. Other forms of analysis are possible, such as recognition of an individual based on gait or heat signature. In principle, "tipping and queuing" could be performed automatically to assist in manual assessment of the data.

7. Pattern-of-life analysis. Many of the tasks listed above assume analysis of single images, or of video data over short time intervals. It should also be possible to perform in-depth processing of image/video data over several days or weeks. Travel patterns (coming and going) could be identified and logged, as could heat signatures of buildings or other forms of infrastructure.

8. Anomaly detection. Related to the previous task, after patterns have been identified, then deviations from those patterns can potentially be identified. Examples of such anomalies include objects appearing in video data at unexpected times, as well as expected objects *not* appearing at expected times.

9. Networked camera. IR cameras are now available that support wireless connectivity, possibly to include remote control and streaming. Some processing of images and video could be performed in real time, if supported by the camera. A possible product to consider is the FLIR E-series, which provides WiFi connectivity and is described at http://www.flir.com/thermography/americas/us/view/?id=56911. Ultrawideband communications might be an alternative to WiFi. A possible UWB subsystem is described at http://www.starixtech.com/index.html.

10. Multiple cooperative cameras. Surveillance and tracking tasks often involve multiple cameras. It would be possible to extend the current integrated system to accommodate several imaging stations.

4. Bibliography

[1] FLIR Inc., "BHM-Series Bi-Ocular Thermal Night Vision Camera Operator's Manual," Document Number 432-BHM0-15-10, Revision 110, March 2013. Available on-line (accessed April 2014): http://support.flir.com/DocDownload/Assets/87/English/432-BHM0-15-10$110.pdf

[2] Android reference material (accessed April 2014): http://developer.android.com

[3] WugFresh, Nexus Root Toolkit v1.8.2 (accessed April 2014): http://www.wugfresh.com/nrt/

[4] XDA Developers, Nexus 4 forum (accessed April 2014): http://forum.xda-developers.com/nexus-4/orig-development/usb-otg-externally-powered-usb-otg-t2181820

[5] Android reference material (accessed April 2014): http://developer.android.com/guide/topics/sensors/sensors_overview.html

[6] Android reference material (accessed April 2014): http://developer.android.com/reference/android/hardware/SensorEvent.html#values

Appendix A: Quick-Start Guide

The following instructions have been written for the Nexus 4 Android device. If using a Nexus 5, the Power Bank is not needed and references to the Power Bank can be ignored.

PREPARE THE SYSTEM FOR OPERATION

Step 1. Review the camera manual to become familiar with normal operation of the camera.

Step 2. Charge the batteries. For the camera, refer to the procedure that is described in the camera manual. Also charge the Android device and the Power Bank.

Step 3. Check the status of sensor logging on the Android device. To do this, run the **Sensor Controller** application. If the status message “Sensor Logging In Progress” appears on the display, then no action is needed. Otherwise, press Start Sensor Logging and verify that “Sensor Logging In Progress” is displayed. When finished with this step, press Dismiss User Interface.

Step 4. Turn on the camera and select the desired options. (The camera manual describes auto-standby mode and other options, for example.) Verify that the camera’s SD memory card is properly inserted.

Step 5. *Important*: Verify that the camera’s date and time settings match those of the Android device. This step is necessary for proper synchronization of camera files with Android sensor data. (If the camera’s date and time need to be set, then you will need to configure the camera from a laptop PC over a USB connection. Refer to Section 2.7 of this document for further discussion.)

Step 6. Physically mount the Android device onto the camera.

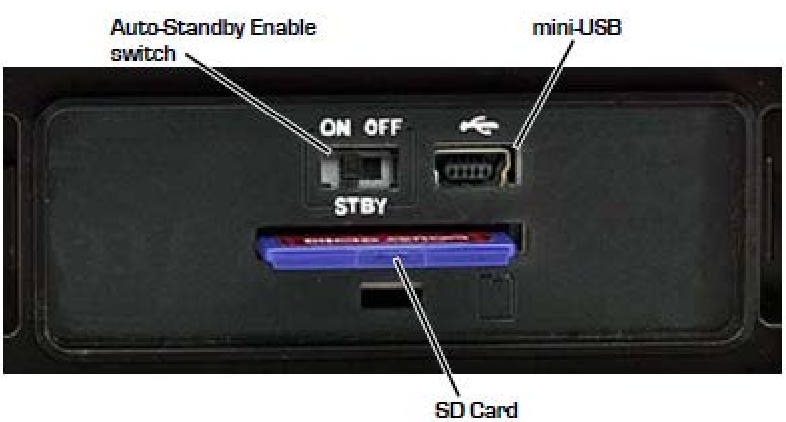
Step 7. Verify that the camera is *not* connected via USB cable to any other device. (A USB connection temporarily disables the camera’s push-button controls.)

ACQUIRE IMAGES AND VIDEOS

Step 8. Capture image and video files by pressing the camera’s pushbutton controls, just as you would for normal operation of the camera.

TRANSFER IMAGE AND VIDEO FILES TO THE ANDROID DEVICE AND PERFORM SYNCHRONIZATION

Step 9. After finishing image/video acquisition, a powered USB connection is needed between the camera and the Android device. Cabling is described in Appendix B of this report. First, connect a male mini-USB connector to the bottom of the camera as shown below; next, provide power to the USB cabling from the Power Bank; last, connect a male micro-USB connector to the Android device.



Step 10. After connecting the camera to the Android device, the Android operating system will recognize that the USB connection has been established. (Up to two minutes may be needed for recognition to occur, according to the camera manual.)

Step 11. Start the **Camera Controller** application on the Android device, unless the app has already launched automatically. Verify that the message "Camera Connected" is displayed.

Step 12. Press the button labeled Start File Transfer. The **Camera Controller** application will now copy all image and video files from the camera's SD card to the memory of the Android device. The **Camera Controller** application will then automatically perform synchronization of the image and video files with data from the Android device's organic sensors that have been logged by the **Sensor Controller**. (Transfer and synchronization may take several minutes, depending on the number and size of the files. More details are provided in Section 2.6 of this manual.)

Step 13. For typical use, press the **Camera Controller** button labeled Delete All Files On Camera, and confirm the deletion. *Caution*: This step will clear the entire SD memory card in the camera.

Step 14. Press the **Camera Controller** button labeled Close.

Step 15. Start the **Sensor Controller**. If finished with data collection activities for the day, press Stop Sensor Logging. If it is desirable to free up storage space on the Android device, also press Delete Log Files and confirm. Then press Dismiss User Interface.

Step 16. The synchronized image/video/sensor files are now ready to be copied or moved from the Android device to another platform for analysis. All of the files are available in the Media folder of the Android device.

Appendix B: Connecting the Camera to the Android Device

This appendix shows the USB interconnections that are needed between the camera and the Android device. These interconnections are needed *only* when the Camera Controller app performs file transfers and synchronization. (The camera's push-button controls are disabled when the USB connection is present.) Figures B.1 and B.2 show the USB arrangements that are needed for the Nexus 4 and the Nexus 5, respectively. For the Nexus 4, the camera's USB port needs to be connected to an external power source *before* being connected to the Android device; otherwise, the Android device will not detect the USB connection.

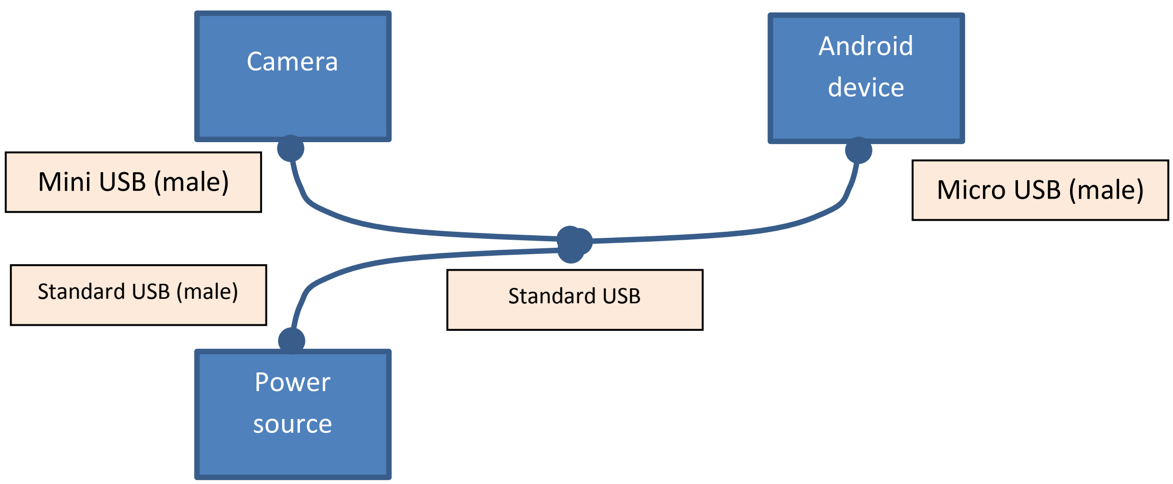


Figure B.1. USB interconnections for the Nexus 4. Notice that an external power source is needed.

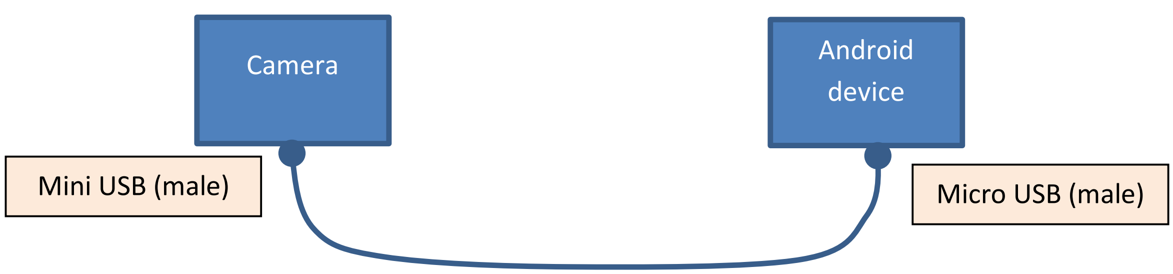


Figure B.2. USB interconnections for the Nexus 5. An external power source is not needed.